New Shuffle Algorithm for Lossless Compression Method of JPEG File

SURAGALI CHANTI, CH RAMESH
Department of CSE, AITAM, Tekalli, Srikakulam, (A.P.), India

***Abstract*— In computing JPEG is a commonly used method of lossy compression for digital photography (image). The degree of compression can be adjusted allowing a selectable trade of between storage size and image quality. JPEG compression is used in a number image file formats. The compression method is usually lossy, meaning that some original image information is lost and cannot be stored. There is an optional lossless mode defined in the JPEG standard. Image of JPEG format has high image quality characteristics, which is widely used in image transmission of computer storage and computer networks. The existing ordinary digital photos mostly use JPEG format. Because the quality of JPEG image is high, it occupied space is relatively large. For a8 million pixel digital camera, its JPEG image is up to 5M bytes. In order to save storage space and reduce the image transmission time, we usually compressed original JPEG image files and then save or transfer them. The compression effect of existing lossless compression algorithm for JPEG file is poor. This paper introduces lossless compression method combined by shuffle algorithm and lossless compression algorithm and proposes a novel algorithm. Combining the new approach and universal lossless compression algorithm we can compress losslessely to JPEG file. With this proposal we can further remove the internal documents redundancy and reduce the file size. There were two existing methods so called lossless compression and lossy compression. The proposed method obeys both the algorithm advantages. The novel approach is called shuffle algorithm which effectively structure the JPEG image file before compression, then combining with common lossless compression algorithm can remove inside information redundancy. Shuffle algorithm is resetting the data bits and arrange bits into another arrangement. There are many shuffle functions, such as uniform shuffle, the first k-th sub-shuffle, the kth ultra shuffle, etc.. Then anti-shuffle function will transform original arrangement into the sequence. The suggested approach is to be developed using mat lab tools for image processing and retrieval techniques.**
***Keywords*—shuffle algorithm, JPEG, lossless compression.**

I. INTRODUCTION

Image of JPEG format has high image quality characteristics, which is widely used in image transmission of computer storage and computer networks. The existing ordinary digital photos mostly use JPEG format. Because the quality of JPEG image is high, it occupied space is relatively large. For a 8 million pixel digital camera, its JPEG image is up to 5M bytes. In order to save storage space and reduce the image transmission time, we usually compressed original JPEG image files and then save or transfer them.There are two kinds of existing compression methods. One is lossless compression. This method regard data as combination of information and redundancy, whose work mechanism is to remove or reduce data redundancy, therefore it don't lead to the loss of original data. The other is lossy compression. This compression not only removes or reduces data redundancy, but also ignores some less important details, which has significant compression ratio. But this compression method could lose some information of original document, leading that the unzipped files have difference with original file. [1] For general lossless compression, their representive algorithm is Huffman algorithm, LZ and its derivative algorithms (LZW, deflate, etc.), whose derivative softwares are WinRAR, WinZip, 7-zip, etc. However, the above softwares compress JPEG image files directly. That is extremely unsatisfactory, and even the compressed file size increases. In fact, the original JPEG image file exist certain redundancy because of its different emission. This redundancy cannot be removed unless the original JPEG image files is structured. How to effectively structure the original JPEG image file has become another mean to improve compression ratio.

II. NOVEL SHUFFLE ALGORITHM

Shuffle algorithm is reseting the data bits and arranges bits into another arrangement. There are many shuffle functions, such as uniform shuffle, the first k-th sub-shuffle, the first k-th ultra-shuffle, etc. Then anti-shuffle function will transform original arrangement into the sequence. [2] Novel algorithm proposed in this paper is defined as the same father shuffle algorithm. i.e., if the card in the front of card A is card C and the card in the front of card B is also card C, the define card C as father card of card A and card B. When shuffling, we should plate card A and B in sequence according with its order. The following is a specific description for the combination of JPEG file and the same father shuffle algorithm.

A. *Shuffle algorithm principle*

Structuring the same father shuffle algorithm to JPEG file includes the following steps:
1) Read JPEG image file by binary mode and group them; Grouping is from the first bit of binary number. Generally, per 8 bits is a group. If not reaching 8 bits, fill 0 in the front of the binary number that is not enough8 bits until reaching 8 bits;
2) Convert each binary code into unsigned decimal number, and the unsigned decimal number converted by the first binary code group is stored in a new array H; Since the unsigned decimal number converted by the first packed data is between 0 and 255. In order to save space, the unsigned decimal number can be set to take up 1 byte space; 3) Establish 256 arraies Ei , i=0-255 orderly for storing the child data back of each father data;For any two adjacent groups, the packet data in front is set to father data and the back is set to child data; *4)* According to the order from small to large, and the child data back of each father data is stored in the i-th array E[i] in step 3), where i is equal to father data. If not existing father data same to it, array E[i] is empty;

5) Determine whether the traversal ends. If it is right, count the length of every traversed array and store them in a new array G; Other wise, continue step 4); The length of each array element in array G occupies 4 bytes spaces.
6) Connect the contents of 256 Ei arraies in turn and store them in a newarray I;
7) Connect array H, array G and array I ends in turn and store them in a new array J. Then array J is the transformed JPEG image file.It is seen that the transformed file is larger 1001 bytes than the original file.

B. *Anti-shuffle algorithm*

Anti-shuffle the transformed JPEG image file, the steps are as
Follwing:
1) Read the reversed transformed JPEG image file by binary to a new array J, and then packet them by data;
2) Transform each data to unsigned decimal number, read the first unsigned decimal number and store them in a new array H; Read the array length elements storing subdata and store those length elements in a new array G in turn; Read the compressed data of stored space orderly; i.e., read the unsigned decimal number occuping 1byte space in array J in array H, and read every 4 bytes below he unsigned decimal number as one element in array G;
3) Establish 256 arries Ei, i = 0-255 orderly for storing child data after the father data corresponding to i ;
4) Store the remained data in array J in array E[i] established in step 3) according to the length of child data array in array G;
5) Establish arrayA and store the data of arrayH in the first position of array A;
6) Read the data of array A orderly as father data, find the data satisfying that i is equal to the data in E[i] of the first not reading data of father data in 256 array group E[i], and store this child data in the empty position below this father data in array A; At the same time, sign the child data in the corresponding array read;
7) Scan 256 arrays Ei, and check whether the flags of all data have read. If not read, continue step 6); otherwise, stop the scan, and store array A as file. This file is the original JPEG image file.
C. *Flow chart is as follows:*
Compression / decompression flow charts are shown in Figure 1 and Figure 2.

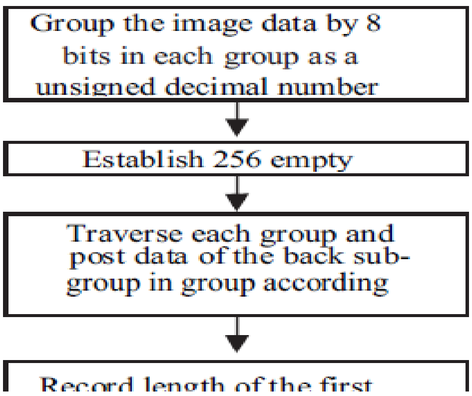

Figure 1: Flow chart of shuffle for JPEG image file

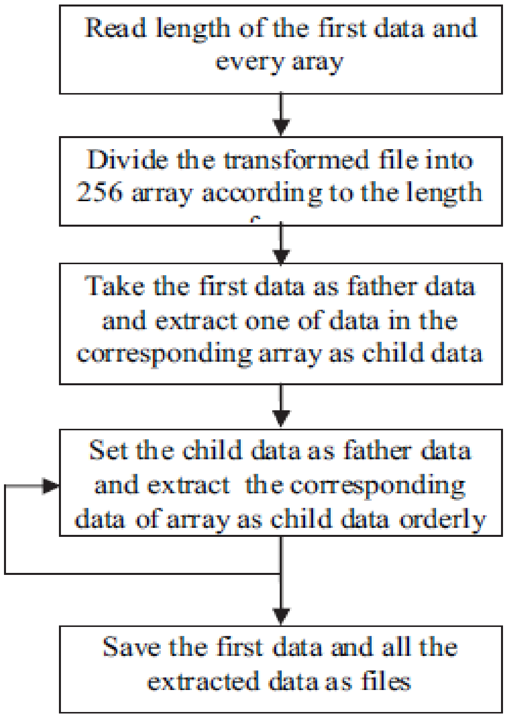

Figure 2: Flow chart of restoring process for decompressed JPEG image file.

III. LOSSLESS COMPRESSION/DECOMPRESSION STEPS BASED ON THE SAME FATHER SHUFFLE ALGORITHM OF PEG FILE

A. *Compression Steps*
1) Reversibly transform the JPEG image file in accordance with the rule of the same father shuffle, then the transformed JPEG image file can be gotten.
2) Compress the transformed JPEG image file using LZ77 coding or other improved algorithm, then the compression is completed.

The foundation of image compression is information theory, as laid down by the likes of Shannon in the late 1940s. Information theory tells us that the information of an event is

$$\log_2 \frac{1}{p(e)} \text{ bits}$$

Where p(e) is the probability of the event occurring. Thus, the information content of an event is directly proportional to our surprise at the event happening. A very unlikely event carries a lot of information, while an event that is very probable carries little information. Encoding an image can be thought of as recording a sequence of events, where each event is the occurrence of a pixel value. If we have no model for an image, we might assume that all pixel values are equally likely. Thus, for a greyscale image with 256 grey levels, we would assume p(e) = 1=256 for all possible pixel values. The apparent information required to record each pixel value is then log2 256 = 8 bits. Clearly, this is no better than the raw data representation mentioned above. However, due to the spatial smoothness common in images, we expect a given pixel to be much like the one before it. If the given pixel value conforms to our expectation of being close to the previous value, then little information is gained by the event of learning the current pixel's value. Consequentially, only a little information need be recorded, so that the decoding process can reconstruct the pixel value. This idea of using previous pixel values to lower the information content of the current pixel's encoding has gone under several names: Deferential Pulse Code Modulation (DPCM), difference mapping and more generally predictive coding. From early work in the 50s and 60s on television signal coding, to modern lossless image compression schemes, predictive coding has been widely used. The common theme has always been to use previous data to predict the current pixel and then only the prediction error (or prediction residual) need be encoded. Predictive coding requires the notion of a current pixel and past pixels and this implies a one-dimensional (1D) sequence of pixels. However, image data is two-dimensional. To correct for this mis-match a 1D path is needed that visits every pixel in the 2D image. By far the most common path is raster-scan ordering, which starts at the top left of an image and works left to right, top to bottom, over the whole image.

B. *Decompression steps*

1) Decompress the compressed JPEG image file using LZ77 encoding or other improved algorithm;

2) Anti-structure the reversible JPEG image file using anti-shuffle algorithm, then the original JPEG image file can be gotten.

IV. EXPERIMENT

Before Experiment environment: Core 2 T9300 computers, windows xp Experiment data: Canon EOS 400D Digital, Cannon PowerShot D630 etc. cameras and online resource pictures.

Experiment tool: gzip common lossless compression Algorithm some test results are shown in Table I:

TABLE I. JPEG WITH WINRAR、WINZIP AND GZIP+NOVEL SHUFFLE ALGORITHM

JPEG bits	Winrar	Winrar Compress ratio	Winzip	WinZip Compress ratio	Gzip+novel shuffle algorithm	Gzip+novel shuffle algorithm compress ratio
3681869	3681869	1	3677638	0.998850855	3638128	0.988119892
3770812	3770812	1	3766598	0.998882469	3726476	0.98824232
3945319	3945319	1	3941177	0.998950148	3899604	0.988412851
3417356	3417356	1	3412817	0.99867178	3377434	0.988317869
3835407	3835407	1	3831230	0.998910937	3789818	0.988113647
4475246	4475246	1	4471177	0.999090776	4419403	0.987521803
4792901	4792901	1	4788787	0.999141647	4729471	0.986765844
5384009	5384009	1	5378453	0.998968055	5301398	0.984656229
5449415	5449415	1	5443992	0.999004847	5368324	0.98511932
4626088	4626088	1	4622023	0.999121288	4564513	0.986689618
3565861	3565861	1	3561495	0.998775611	3520582	0.987302085

V. RESULT ANALYSIS

After Volume analysis: The above results are only partial data.

Many experiment data show that this method can carry on reduced and lossless compression to JPEG file and the compression result of JPEG image data whose size is above 1M bits is better than that of compression software. Real-time analysis: In the above experiment environment, for compressing a single 20M bits JPEG file. Winara takes about 14 seconds, winzip takes about 3seconds and the same father shuffle + gzip lossless compression method takes about 5.5 seconds. For the data in the above table, there are 40 JPEG image file. Winrar compression takes 72 seconds, winzip compression takes 12 seconds and the same father shuffle + gzip lossless compression method takes about 40 seconds.

VI. CONCLUSION

In current, JPEG image is popular and widely used in various occasions, whose volume becomes the restriction of transmission and savage along the development of digital devices. This paper studies how to reduce size as possible in the case of not loss of quality. And this paper proposes a new shuffle algorithm. Combining lossless compression algorithm from another point, we give a lossless compression method of JPEG image file. Experiment shows that this method has real-time and its compression rate is higher 1-3% than that of WinRAR, WinZip, 7zip and other popular softwares. The specific theory support is under investigation

REFERENCES

[1] Ternary Tree & FGK Huffman Coding Technique Dr. Pushpa R.Suri † and Madhu Goel Department of Computer Science & Applications, Kurukshetra University, Kurukshetra, India

[2] Massachusetts Institute of Technology Department of Electrical Engineering and Computer Science.

[3] *Compression Using Fractional Fourier Transform* A Thesis Submitted in the partial fulfillment of requirement for the award of the degree of *Master of Engineering In Electronics and Communication. By Parvinder Kaur.*

[4] RL-Huffman Encoding for Test Compression and Power Reduction in Scan Applications-MEHRDAD NOURANI and MOHAMMAD H. TEHRANIPOUR, the University of Texas at Dallas.

[5] A Novel VLSI Architecture of Hybrid Image Compression Model based on Reversible Blockade Transform C. Hemasundara Rao, *Student Member, IEEE,* M. Madhavi Latha, *Member, IEEE*

Mr. Suragali Chanti received the MCA master Degree from Andhra University, vishakapatnam in 2006 and He is currently pursuing M.Tech in the Department Of Computer Science and Engineering, AITAM, Tekalli, JNTUniversity. His research interests include Image Processing and Computer Networks.

Ch. Ramesh received the B. Tech degree from Nagarjuna University, Nagarjuna Nagar, India and the M. Tech degree in Remote Sensing from Andhra University, Vishakhapatnam, and another M. Tech degree in Computer Science & Engineering from JNTU, Hyderabad. He is currently pursuing the Ph. D degree in Computer Science from the University of JNTU, Hyderabad. He is a professor in the department of Computer Science & Engineering, AITAM, Tekkali, India. His research interests include Image Processing, Pattern Relegation, and Formal Languages.